

SPECIFICATION AMENDMENTS:

Please replace the paragraph from page 1, line 24 through page 2, line 6, with the following amended paragraph:

Next, a process in an encoder, described subsequently, for creating an ECC block 30 by a plurality data sectors is explained in FIG. 1B. As shown, an ECC block is formed by 16 data sectors is explained in FIG. 1B. To forming the ECC format, each data sector 20 including ID 21, IED 22, RSV 23, main data 24, and EDC, each data sector having 2064 bytes arranged in an array of 12 data rows each containing 172 bytes. The first data row should start with three fields: ID, IED, and RSV, followed by 160 bytes main data. The next 10 data rows should each ~~contains~~ contain 172 bytes main data, and the last data row should contain 168 bytes main data followed by 4 bytes EDC.

Please replace the paragraph on page 5, lines 15 through 30, with the following amended paragraph:

FIG. 4 is a flowchart showing conventional DVD disc encoding. First,

main data is read from the host computer 14 through the interface (IDE Bus) 13 shown in FIG. 3 and written to the DRAM 10 (S1). Next, main data restored in the DRAM 10 is read (S2). Next, the 2-byte ID Error Detection code (IED) is generated to correct errors in the 4-byte ID information (S3). Next, 6 bytes of reserve data (RSV) denoting copyright is generated (S4). Next, 4 bytes of error detection code (EDC) is generated for detecting errors (S5). Next, main data the data including main data, ID, EC, RSV, and EDC is scrambled (S6). Thus, a data sector is obtained. Next, 10 bytes PI is generated according to the scrambled data, and is attached to the 16 data sectors ~~are attached to the 10 bytes PI~~ (S7). Next, the scrambled data, ID, IED, RSV, EDC and PI are stored to the DRAM (S8). The data stored in the DRAM is read again to generate 16 bytes PO and interleave the data sectors and PO (S9), ~~which are then~~ Next, the 16 data sectors interleaving the 16 bytes PO are stored in the DRAM (S10). Thus, the data stored in the DRAM is read to be written to the disc (S11).

Please replace the paragraph on page 6, lines 15 through 21, with the following amended paragraph:

Also, it is noticed that in order to generate PI (in step S7 and PO (in step

S9), main data must be scrambled in advance (in step S6), the scrambled data would be stored in DRAM; as a result if there ~~[[has]]~~ is a block that needs the same main data to record on to the same disc, thus the scrambled data in DRAM should be read out and descrambled to the original main data and then scrambled again due to the ID ~~change~~ information being changed.

Please replace the paragraph from page 6, line 28 through page 7, line 5, with the following amended paragraph:

The present invention provides a method for generating recording data, including vertically scrambling main data stored in DRAM to generate PO, wherein the generated scrambled main data is not stored back into DRAM and ~~remain~~ the main data in DRAM remains the same; scrambling the main data again to generate PI; and delivering the scrambled data ~~[[in]]~~ accompanied with ID, IED, RSV, EDC, PI, and PO ~~arrange in orders~~ in proper sequence for recording onto a disc.

Please replace the paragraph on page 7, lines 6 through 9, with the following amended paragraph:

As per the description above, because the scrambled data ~~while used for~~ generating PO is ~~not stored back into~~ written over the main data stored in DRAM, as a result, ~~the embodiment provides a same main data since the main data~~ stored in DRAM ~~are not scrambled or changed~~ remains unchanged.

Please replace the paragraph on page 8, lines 6 through 19, with the following amended paragraph:

Please refer to FIG. 5, which shows an operating flow illustrative of the ~~[[processor]]~~ process of the scramble method for generating recording data according to the present invention. Firstly, the main data of a block stored in DRAM is accessed in step 510 for performing a vertically scrambling procedure on vertically-sequenced data sectors of the main data and deriving ~~such that an ECC~~ outer-code parity (i.e. PO) from the result ~~is derived in (step 520)~~. The scrambled data used for generating PO is not stored back into DRAM (i.e. data in the DRAM is not scrambled data). That is to say the main data in DRAM still remains

unchanged. Therefore Next, a horizontally scrambling procedure is then activated to scramble the main data again, and to derive inner-code parity (i.e. PI) of each data sector from the result and outer-code parity so as to generate PI required by the currently processed block in (step 530). Then, the scrambled data [[in]] accompanied with ID, IED, RSV, EDC, PI, and PO are arranged in orders proper sequence and ready to be recorded onto a disc in step 540.

Please replace the paragraph on page 11, lines 3 through 21, with the following amended paragraph:

FIG. 7 shows an example of 5 ECC blocks 7A~7E in a buffer. as know, a common used method when recording data is: firstly, writing ID into each ECC blocks in orders; then writing data read from DRAM into each ECC blocks in orders (i.e. the first data with ID 0 is firstly written into the first block 7A, then the second data with ID 1 is then written into the second block 7B, and so no.). Hen a block is filled with data, the scrambling method of the invention described above is performed, and then the scrambled data with corresponding ID, IED, RSV, EDC, PI, and PO are ~~reecording~~ recorded onto a disc in sequence. It is noted that, writing data into blocks is simultaneous with recording data onto the disc, but under the

same time, the writing block could be unequal to the recording block; that is to say, when data are continuously written into a block, the other blocks which ~~[[data]] have been written into is recording~~ encoded are recorded onto the disc at one time (i.e. assuming the writing block is 7C, so that the recording block could be 7A or 7B which is not limited). Furthermore after data with ID 4 is written into the last block 7E, the next data with ID 5 is written into the first block 7A and so on.

Please replace the paragraph from page 11, line 22 through page 12, line 2, with the following amended paragraph:

It is observed that, a ~~common~~ commonly used method when writing the same data into different blocks is descrambling the scrambled data to original data, scrambling the original data as another scrambled data and then written into the desired block. Under the condition described above, memory bandwidth is

wasted. As a result, the present invention provides a method when recording the same data onto different blocks. Because the data in the block is ~~[[no]]~~ not scrambled data due to generating PO in the present invention (the scrambled data for generating PO is not stored in DRAM), so that, if the same data is written into different blocks, the same data could be directly derived from DRAM without descrambling.